Objectives

- Analyzing proofs
- Introduction to problem solving
  - Our process, through an example

4 p.m. – Alicia Grubb, faculty candidate talk, P405
3:30 p.m. reception

Wiki:
- Everyone log in okay?
- Decide on either using a blog or wiki-style journal?

Review

- What are our goals in solving problems?
- How do we show that our solutions are correct and efficient?
- What proof techniques did we discuss?

Proof Summary

- Need to ***prove*** conjectures
- Common types of proofs
  - Direct proofs
  - Contradiction
  - Induction
- Common error: not checking/proving assumptions
  - "Jumps" in logic

Process, through example

INTRODUCTION TO PROBLEM SOLVING

Matching Residents to Hospitals

- **Goal**: Given a set of preferences among hospitals and medical school students, design a *self-reinforcing* admissions process.

- Applicant *a* and hospital *h* are ***unstable*** if
  - *a* prefers *h* to its assigned hospital
  - *h* prefers *a* to one of its admitted students

- **Stable assignment**: Assignment with no unstable pairs
  - No incentive for some pair of participants to undermine assignment by joint action
    - Unstable pair could each improve their situation by swapping with current assignment
    - Self-reinforcing

What details make this problem tricky?
What info do we need to solve problem?

Stable Matching Problem

Simplified version of resident-matching problem

- **Goal**: Given *n* men and *n* women, find a "suitable" matching
  - Participants rank members of opposite sex
  - Each man ranks women in order of preference
  - Each woman ranks men in order of preference

favorite ↓ least favorite ↓

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite ↓ least favorite ↓

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Stable Matching Goals

- **Perfect matching**: everyone is matched monogamously
  - Each man is paired with exactly one woman
  - Each woman is paired with exactly one man
- **Stability**: no incentive for some pair of participants to undermine assignment by joint action
  - An *unmatched* pair *m*-*w* is **unstable** if man *m* and woman *w* prefer each other to current partners
  - Unstable pair *m*-*w* could each improve by eloping
- **Stable matching**: perfect matching with no unstable pairs

Stable matching problem:
Given the preference lists of *n* men and *n* women, find a stable matching if one exists.

Analyzing Stability

Instable: *m* prefers *w* to his woman; *w* prefers *m* to her man

- Is pairing X-C, Y-B, Z-A stable?

favorite → least favorite

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite → least favorite

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Analyzing Stability

- Is pairing X-C, Y-B, Z-A stable?
- No. Bertha and Xavier prefer each other

favorite → least favorite

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite → least favorite

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Stable Matching Problem

Instable: *m* prefers *w* to his woman; *w* prefers *m* to her man

- Is pairing X-A, Y-B, Z-C stable?
- Yes.

favorite → least favorite

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite → least favorite

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Any Questions?

- What are you wondering about this problem/its solution at this point?

Any Questions?

- What are you wondering about this problem/its solution at this point?
- Hopefully:
  - Is there a stable matching for every pair of preference lists?
  - If so, is there an algorithm to find the stable matching?
  - Can we be fair in the matching? (preferences)
  - Will the matching always be the same?

Thoughts on Solving Problem

- What do we need to solve the problem?
- What do we know?
- Where should the state start?
- What are some initial ideas about approaches?

Thoughts on Solving Problem

- Initially, no one is matched
- Pick an arbitrary man and have him match with his favorite woman.
  - Are we guaranteed that pair will be part of a stable matching?
- Should a woman accept her first offer? If not, what should she do?
- When are we done? Do we need to consider all combinations?

Propose-And-Reject Algorithm

[Gale-Shapley 1962]

- Intuitive method that guarantees finding a stable matching

```
Initialize each person to be free
while some man is free and hasn't proposed to every woman
    Choose such a man m
    w = 1st woman on m's list to whom m has not yet proposed
    if w is free
        assign m and w to be engaged
    else if w prefers m to her fiancé m'
        assign m and w to be engaged and m' to be free
    else
        w rejects m
```


Applying the Algorithm

favorite ↓ least favorite ↓

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite ↓ least favorite ↓

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

```
Initialize each person to be free
while some man is free and hasn't proposed to every woman
    Choose such a man m
    w = 1st woman on m's list to whom m has not yet proposed
    if w is free
        assign m and w to be engaged
    else if w prefers m to her fiancé m'
        assign m and w to be engaged and m' to be free
    else
        w rejects m
```

Applying the Algorithm

favorite ↓ least favorite ↓

	1st	2nd	3rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

favorite ↓ least favorite ↓

	1st	2nd	3rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

```
Initialize each person to be free
while some man is free and hasn't proposed to every woman
    Choose such a man m
    w = 1st woman on m's list to whom m has not yet proposed
    if w is free
        assign m and w to be engaged
    else if w prefers m to her fiancé m'
        assign m and w to be engaged and m' to be free
    else
        w rejects m
```

Observations about the Algorithm

- What can we say about any woman's partner during the execution of the algorithm?
- How does a woman's state change over the execution of the algorithm?
- What can we say about a man's partner?

Observations about the Algorithm

- What can we say about any woman's partner during the execution of the algorithm?
  - Observation 1. He gets "better" → she prefers him over her last partner
- How does a woman's state change over the execution of the algorithm?
  - Observation 2. Once a woman is matched, she never becomes unmatched; she only "trades up"
- What can we say about a man's partner?
  - Observation 3. She gets "worse"

Proving Correctness

- Need to show
  - Algorithm terminates 
  - Result is a perfect matching
  - Result is a stable matching

1) Algorithm Termination

[Gale-Shapley 1962]

Does algorithm terminate?

```
Initialize each person to be free
while (some man is free and hasn't proposed to every woman)
    Choose such a man m
    w = 1st woman on m's list to whom m has not yet proposed
    if w is free
        assign m and w to be engaged
    else if w prefers m to her fiancé m'
        assign m and w to be engaged and m' to be free
    else
        w rejects m
```


Proof of Correctness: Termination

- Claim. Algorithm terminates after at most n^2 iterations of while loop.
  - Hint: How wouldn't the algorithm terminate?

Proof of Correctness: Termination

- Claim. Algorithm terminates after at most n^2 iterations of while loop.
- Pf. Each time through the while loop, a man proposes to a new woman. There are only n^2 possible proposals.

Number of proposals is a good measure for termination
→ strictly increases; limited

Proof of Correctness: Termination

- Claim. Algorithm terminates after at most n^2 iterations of while loop.
- Pf. Each time through the while loop, a man proposes to a new woman. There are only n^2 possible proposals.

Note: not yet discussing the cost *in the body* of the while loop

2) Algorithm Analysis: Perfect Matching

Prove that final matching is a *perfect* matching

- **Perfect matching**: everyone is matched monogamously
- Hint: in algorithm, we know if *m* is free at some point in the execution of the algorithm, then there is a woman to whom he has not yet proposed.

Proof of Correctness: Perfection

- Claim. All men and women get matched.
- Pf. (by contradiction)
  - Where should we start?

Suppose that some man *m* is not matched upon termination of algorithm

Proof of Correctness: Perfection

- Claim. All men and women get matched.
- Pf. (by contradiction)
  - Suppose that *m* is not matched upon termination of algorithm
  - Then some woman, say *w*, is not matched upon termination.
  - By Observation 2, *w* was never proposed to.
  - But, last man proposed to everyone, since he ends up unmatched
    - (by the while loop's condition)
  - Contradiction ▪

Assignments

- Review Chapter 1
- Journal due Monday/Tuesday (because of MLK day)
  - Preface, Chapter 1.1
  - Check out the content requirements for the journal entries